

Arduino Programming Part 1

Overview

Arduino Environment

Basic code components

- ❖ Two required functions: `setup()` and `loop()`
- ❖ Variables
- ❖ Calling built-in functions

References

These notes borrow from

- ❖ Arduino web site
 - ▶ <http://arduino.cc/en/Guide/Environment>
 - ▶ <http://arduino.cc/en/Tutorial/HomePage>
- ❖ Adafruit tutorial #1 and 2
 - ▶ <http://www.ladyada.net/learn/arduino/lesson2.html>
- ❖ Leah Buechley's Introduction to Arduino
 - ▶ http://web.media.mit.edu/~leah/LilyPad/03_arduino_intro.html

Arduino Web Site References

Overview of the development environment

- ❖ <http://www.arduino.cc/en/Guide/Environment>
- ❖ <https://learn.sparkfun.com/tutorials/redboard-hookup-guide>

Language reference

- ❖ <http://arduino.cc/en/Reference/HomePage>

Code tutorials

- ❖ <http://arduino.cc/en/Tutorial/HomePage>

Basic Process

Design the circuit:

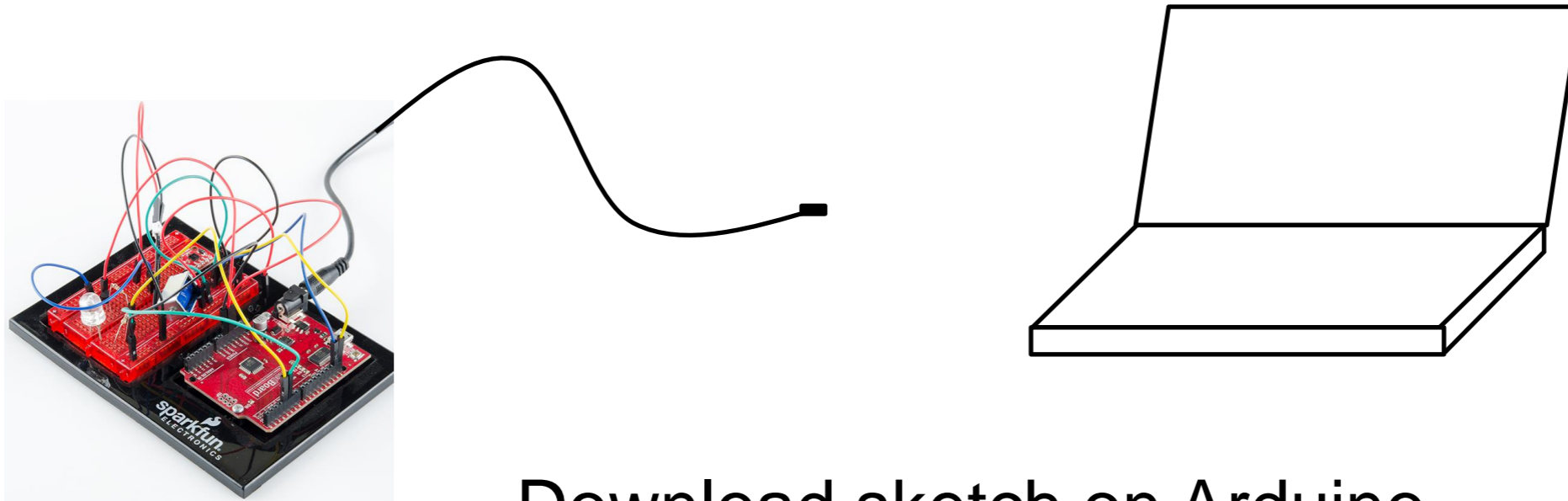
- ❖ What are electrical requirements of the sensors or actuators?
- ❖ Identify analog inputs (sensors)
- ❖ Identify digital inputs & outputs (buttons, LEDs, relays)

Write the code

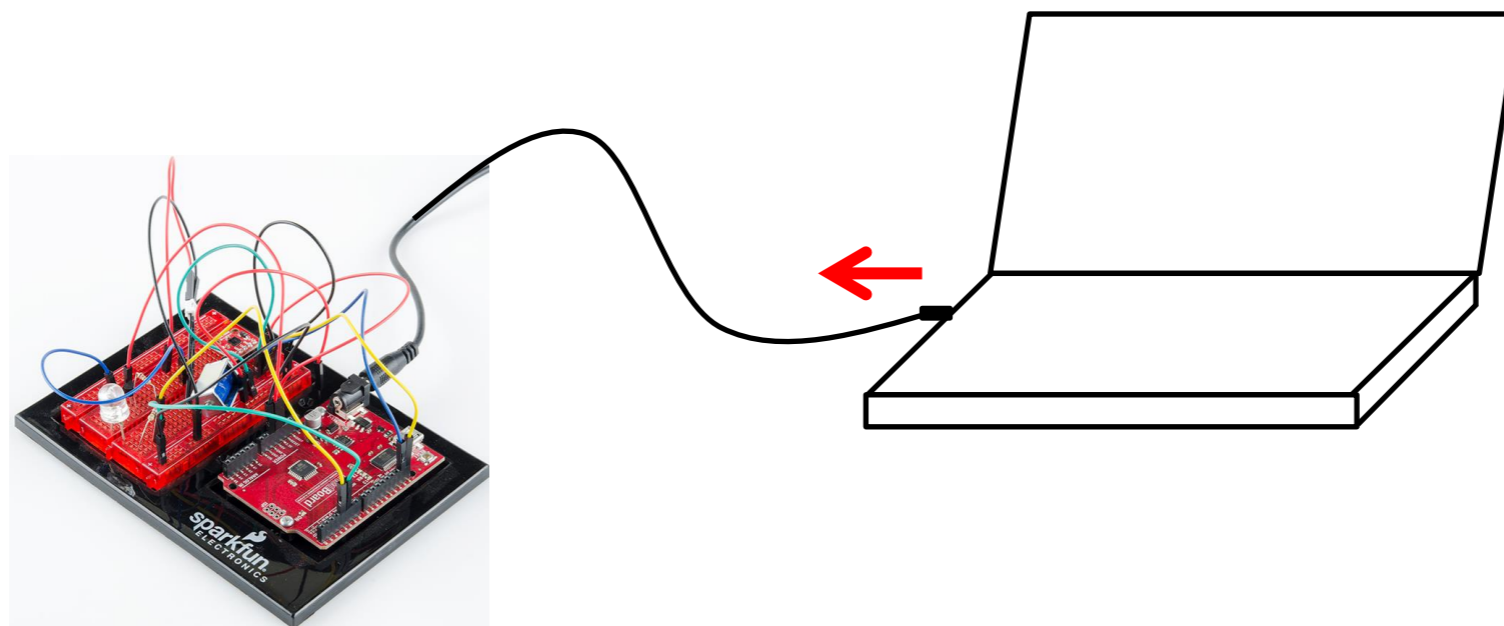
- ❖ Build incrementally
 - ▶ Get the simplest piece to work first
 - ▶ Add complexity and test at each stage
 - ▶ Save and Backup frequently
- ❖ Use variables, not constants
- ❖ Comment liberally

Writing and Downloading Code

Write sketch on PC

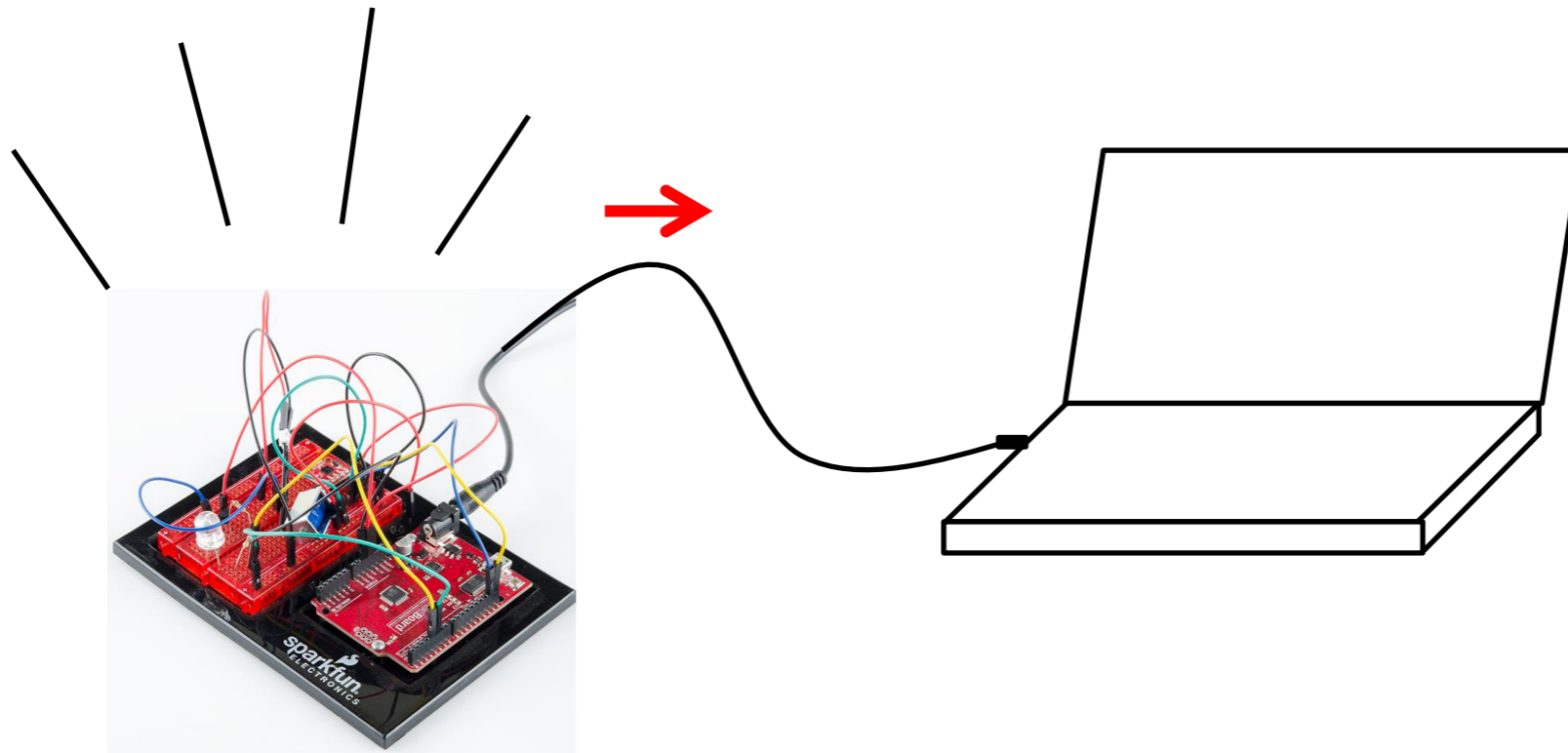


Download sketch on Arduino



Running Code While Tethered

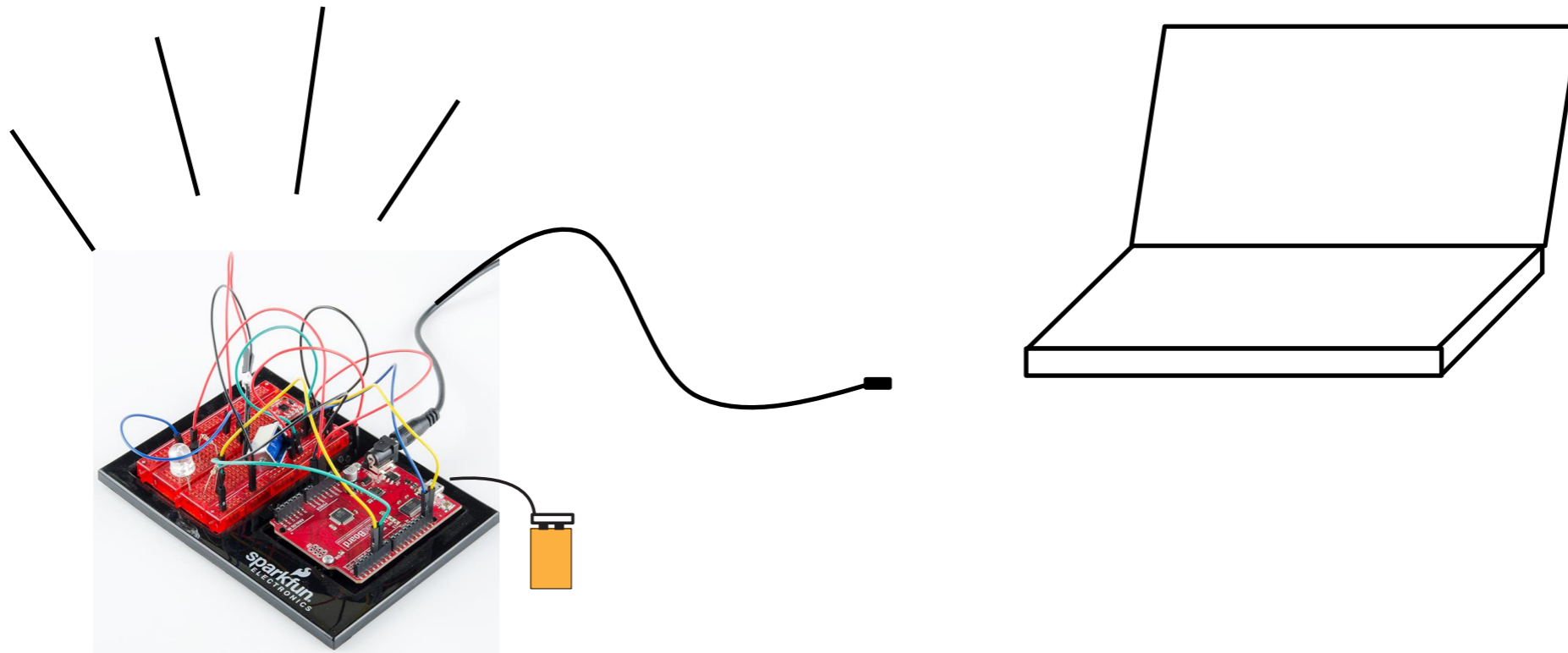
Run sketch on Arduino and send data back to PC



Arduino interacts with
its environment

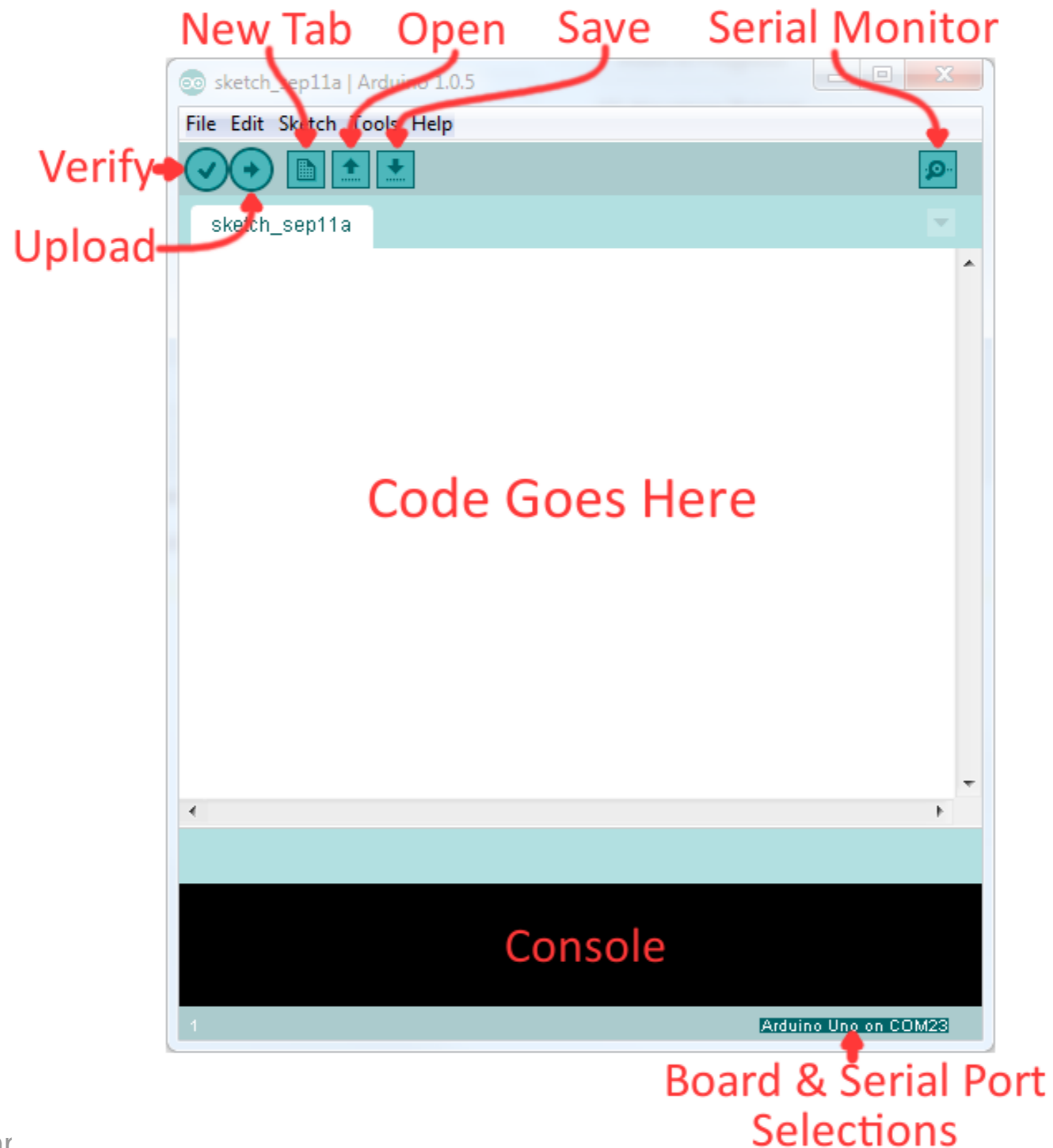
Running Code Stand-Alone

Run Arduino in stand alone mode



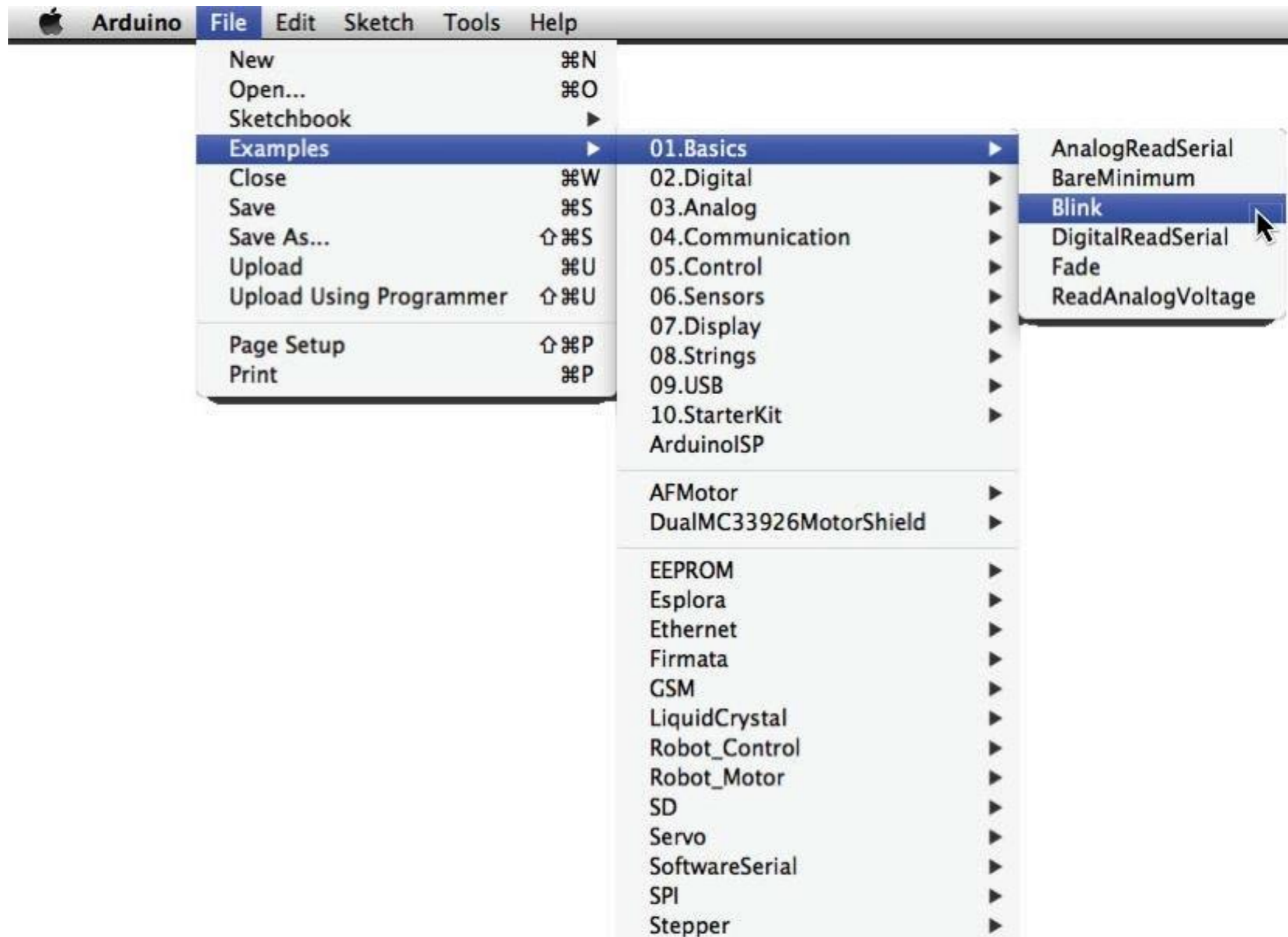
Arduino interacts with its environment and runs on battery power

When you open up the Arduino software

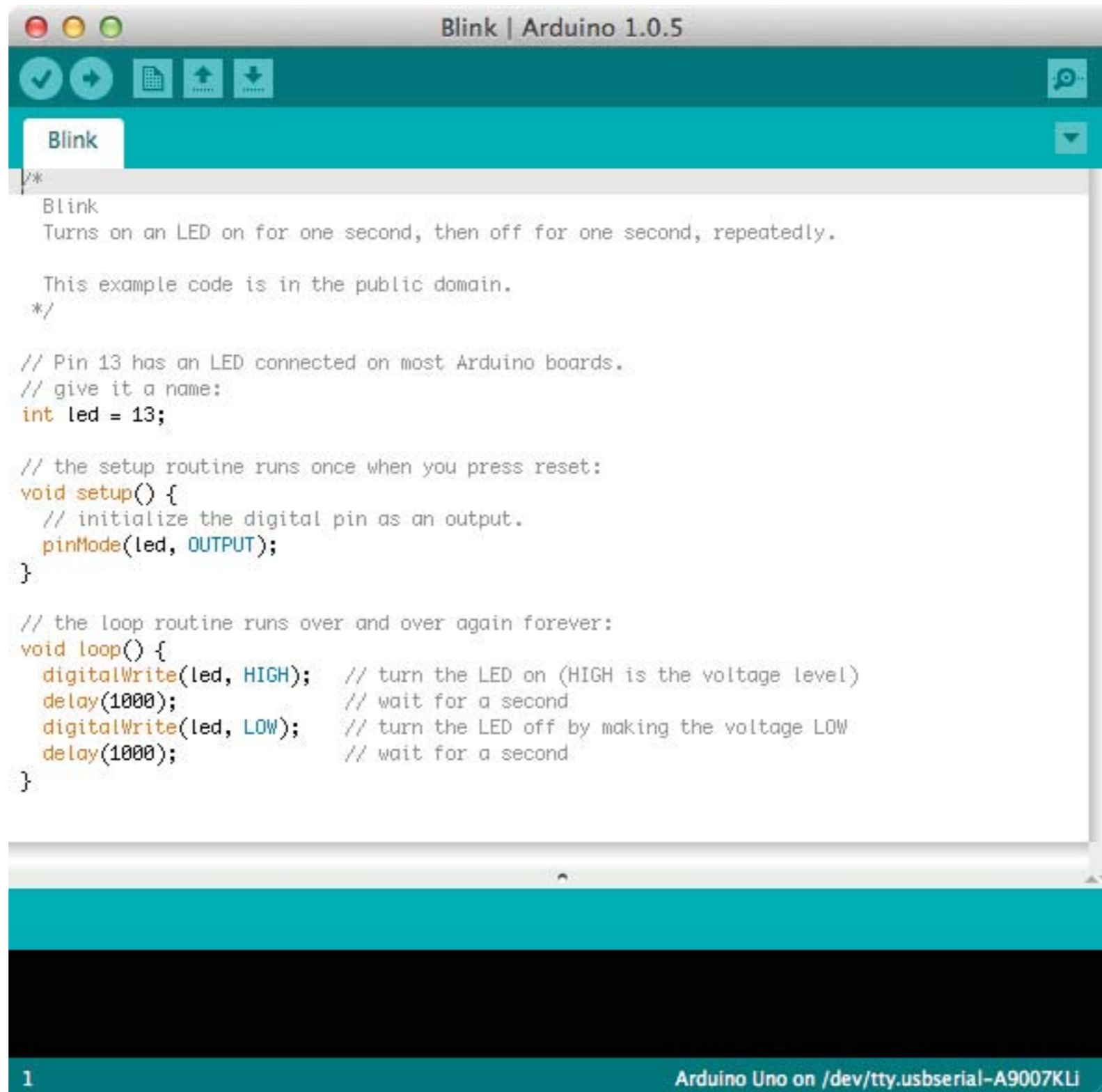


Open the example sketch,
blink.ino

Load “Blink” from the built-in examples



Arduino IDE

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0.5". The top toolbar contains icons for check, run, file, upload, and download. A tab labeled "Blink" is active. The main text area contains the following code:

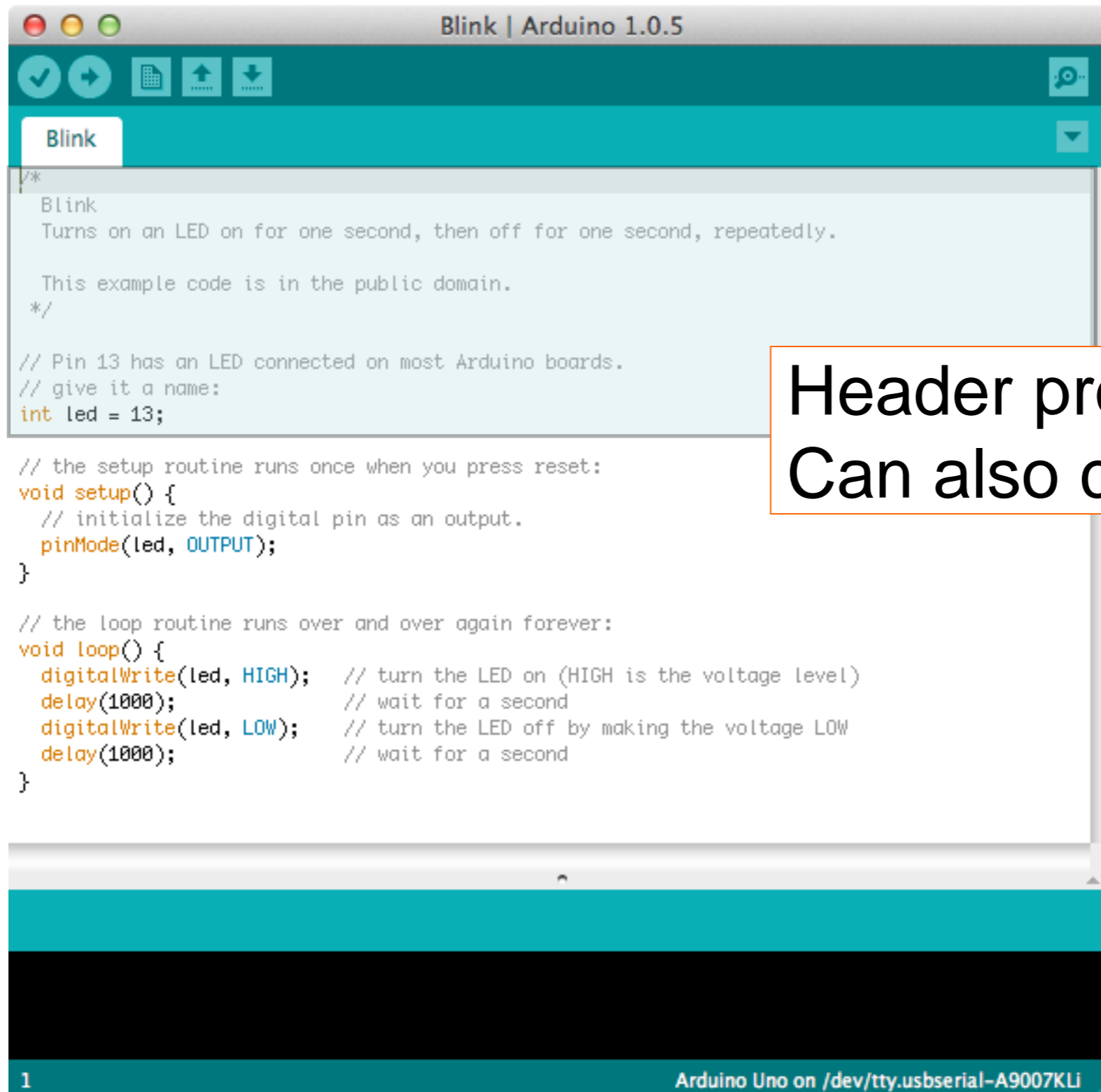
```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 *  
 * This example code is in the public domain.  
 */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

The bottom status bar shows "1" on the left and "Arduino Uno on /dev/tty.usbserial-A9007KLI" on the right.

IDE = Integrated Development Environment

Common Code Structure

Code Structure: Header



```
Blink | Arduino 1.0.5
Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

1 Arduino Uno on /dev/tty.usbserial-A9007KLI
```

Header provides information.
Can also contain code

Code Structure: setup function



```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

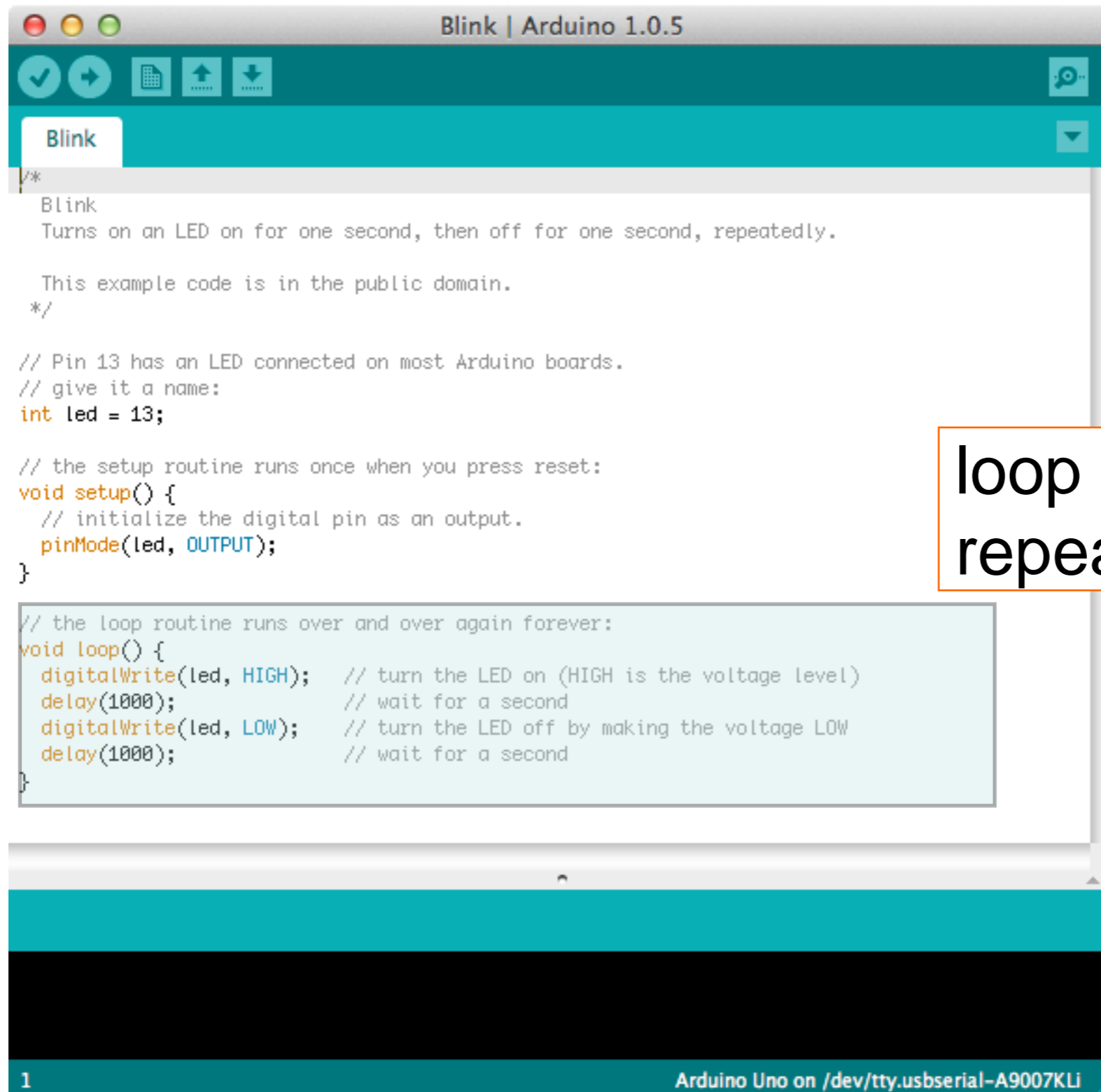
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

setup function is executed only once at the start

Code Structure: loop function



```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

loop function is
repeated indefinitely

Details of the Blink Code

Code

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

`int led = 13;`
creates a variable named “led”
and stores 13 in that variable

`pinMode(led, Output)`
prepare pin number “led”
for outputs of voltage
“led” is a variable

Code

```
Blink | Arduino 1.0.5
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

1 Arduino Uno on /dev/tty.usbserial-A9007KLi
```

`digitalWrite(led, HIGH)`
Sets pin “led” to a value that means the voltage is “on”

`delay(1000);`
tells microcontroller to do nothing for 1000 ms = 1 s

Code

```
Blink | Arduino 1.0.5
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

1 Arduino Uno on /dev/tty.usbserial-A9007KLi
```

`digitalWrite(led, LOW)`
Sets pin “led” to a value that means the voltage is “off”

`delay(1000);`
tells microcontroller to do nothing for 1000 ms = 1 s